

# A KNOWLEDGE BASED APPROACH FOR AUTOMATIC DATABASE NORMALIZATION

**G.Sunitha, Dr.A.Jaya**

**Abstract-** Normalization is a process of analyzing the given relational schemas based on the functional dependencies and using primary key to achieve minimum data redundancy. Normalization is one of the key issues to be carried out manually in database design. This research work aims to resolve this issue by doing normalization of multiple databases automatically.

This approach provides automatic normalization of databases up to 3NF. The unique feature of the research work is automatic normalization and thereby saving time and reducing mind work.

**Index Terms**—First Normal form, Second Normal form, Third Normal form.

## I. INTRODUCTION

For developing any software system, the database normalization helps to avoid data redundancy. If the relational database is used, it consumes time. If we missed out any constraint, we will face problems. While when we automate database, it is easy for normalizing data.

The essence of data normalization is to split your data into several tables that will be connected to each other based on the data within them. By designing database tables carefully, we save space, minimize duplication, protect the data to ensure its consistency, and provide faster transactions by sending less data.

Database normalization is the process of transforming data into well- formed or natural groupings such that one fact is stored in one place. Normalization generally simplifies the relations and reduces the danger of anomalies that may otherwise occur during manipulation of the relations in a relational database. Thus, the normalization procedure provides database

designers with a formal framework for analyzing relation based on their keys and on the functional dependencies among their attributes.

A database management system (DBMS) is computer software that designed for creating and maintaining databases and allows users to retrieve information from that database. MS Access, FoxPro, Dbase are some of the examples for database management system.

Relational database management system (RDBMS) is a type of DBMS in which the database is organized and accessed according to the relationships between data values. It is based on the relational model. The examples for relational database model are Oracle, DB2 and MS SQL Server. The database has to be maintained to avoid redundancy, to eliminate duplicate data. Now a day all the normalization processes for tables are carried out manually. The normal form of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized. Three normal forms called first (1NF), second (2NF), and third (3NF) normal forms were initially proposed.

This section1.0 describes the introduction of the proposed work. Section2.0 focuses on literature survey. Section 3.0 describes the proposed system for automatic normalization. Section 4.0 focuses on the result of the proposed system.

## II. LITERATURE SURVEY

This section focus on literature survey of the research work.

Sherry verma, "Comparing manual and automatic normalization techniques for relational database", [1] proposed the Comparing manual and automatic normalization techniques for relational database, based on the dependency matrix and approach primary key to generate automatically identified the final table.

RadhaKrishna, Rajuvelpula,v. sravya,"A web based relational database design tool to perform normalization", [2] the authors discussed a web based relational database design tool to perform normalization. To developed an interactive relational database design tool to make the normalization process.

Amir Hassan bahmani, MahmoudNaghibzadeh, "Automatic database normalization and primary key generation", [3] the authors proposed an approach for automatic database normalization and primary key generation. In discussed an automatic distinguish one primary key for every final table which is generated. The problem is to normalize the database tables automatically. In the current normalization process, even first normal form, second normal form and third normal forms are difficult by doing automatically.

Abbasifard, Rahgozar, Pournemati, "Using automated database reverse engineering for database integration", [4] the authors approach for automated DBRE and data normalization for integrating systems. Along with normalization process we determine constraint rules. The suggested model provides a schema with complete semantics of legacy databases.

Akehrst, Bordbar, Rodgers, "Automatic normalization via meta modeling", [5] the authors proposed an approach automatic normalization via meta modeling, by using a declarative approach to the specification of the normalization rules and a precisely defined transformation, over a meta-model of a database system design language.

## III. PROPOSED SYSTEM FOR AUTOMATIC NORMALIZATION

The figure 1 shows the architecture diagram for automatic database normalization.

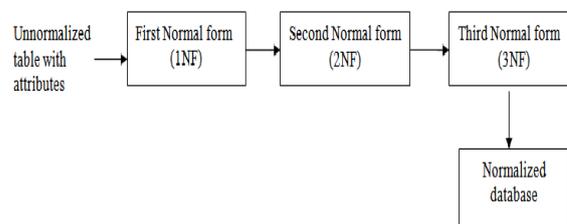


Figure 1 architectural diagram for automatic database Normalization

The components involved in the system and the integration of components that work simultaneously. The proposed system is to make the normalization process automatically by giving some kind of human knowledge to the system to do the normalization process in an effective way. To overcome the limitations of existing system, knowledge based approach is proposed to automate the normalization process of the tables. The goal of proposed system is to do the normalization processes up to third normal form with inclusion of all promising constraints that are needed to make tables as perfect.

- Easy to maintain the record, and identified the record easily.
- The database has to be maintained and avoid redundancy, to eliminate duplicate data.

### First Normal Form

Figure 2 shows the flow chart for 1NF. During the first normalization's process, each and every fields of table are examined whether it has atomic values or composite values in every record. If null value occurs in table then it would be removed from table by entering corresponding data type value. The redundancy of records is also monitored by the implementation of suitable structural query language. If the replicas of records are found then those records are smoothly removed by this system.

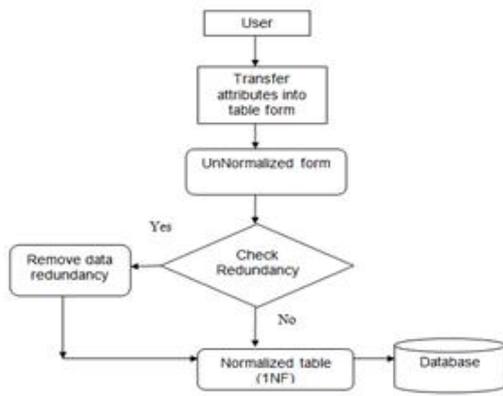


Figure 2 Flow chart for 1NF

table. So in the second normalization part, partial dependency is going to be removed.

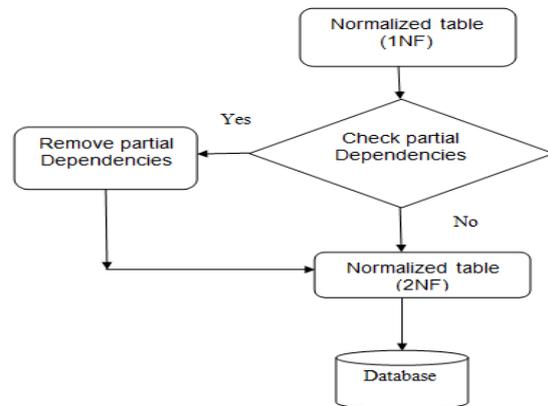


Figure 3 Flow chart for 2NF

stud\_info\_tb

FIELD	TYPE	NULL VALUE	KEY
Studno	Number	NOT NULL	Primary key
Studname	Text	NULL	
course	Text	NULL	
depno	number	NULL	
depname	text	NULL	
subjcode	Number	NULL	
subjname	Text	NULL	

proj\_info\_tb

FIELD	TYPE	NULL VALUE	KEY
projectcode	Number	Not Null	Primary key
projecttitle	Text	Null	
hours	Number	Null	
studno	Number	Null	
studname	Text	Null	
deptno	Number	Null	
deptname	text	Null	

**Third Normal Form**

Figure 4 shows the diagram for third normal form. The third normalization part to satisfy the transitive dependency is removed from the table by extracting the independent columns in the table. It checks the corresponding and all the required constraints in the flow of streams.

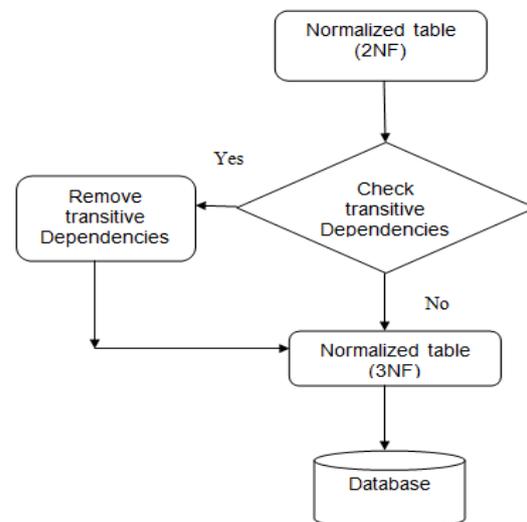


Figure 4 Flow chart for 3NF

**Second Normal Form**

Figure 3 shows the diagram for second normal form. The second normalization part to satisfy the functional dependency. The functional dependency is defined, as every field in a table should be connected with primary key or candidate key of that table. . If any of the non-key fields is connected or logically grouped with a part of candidate key or a part of primary key then it is known as partial functional dependency. The partial dependency should not occur in the

projectcode	projectile	hours	studentno	studentname	deptcode	deptname
12	pensions system	16	96	kala	6	MCA
12	pensions system	18	107	kumar	6	MCA
12	pensions system	15	82	roja	6	MCA
15	salaries system	23	98	kamal	8	MBA
15	salaries system	23	96	kala	6	MCA
15	salaries system	19	102	ramesh	3	IEEE
18	bank system	17	82	roja	6	MCA
18	bank system	15	102	ramesh	3	IEEE

stidno	stidname	course	depno	depname	subjectcode	subjectname
101	mary	Bsc	2	CA	601	comp arch
102	reena	Bsc	2	CA	601	comp arch
103	sudha	Bsc	2	CA	601	comp arch
104	sasi	Bcom	3	HR	604	financial
102	reena	Bsc	2	CA	604	financial
101	mary	Bsc	2	CA	602	human manag
104	sasi	Bcom	3	HR	602	human manag

Figure 5 Unnormalized Table

**IV.RESULTS**

This research work automates the normalization process that leads to time efficiency. If the result contains more than one table, all these normalized tables are automatically connected with main table. This research work is developed fully with user interface that is the interaction between system and user who develops the system. The given table is normalized and if required the tables are split into two or more tables and foreign key is automatically assigned to tables to create relationships among the normalized tables.

projectcode	projectile
12	pension system
15	salaries system
18	bank system

projectcode	hours	studentno	studentname	deptcode	deptname
12	16	96	kala	6	MCA
12	18	107	kumar	6	MCA
12	15	82	roja	6	MCA
15	23	98	kamal	8	MBA
15	23	96	kala	6	MCA
15	19	102	ramesh	3	IEEE
18	17	82	roja	6	MCA
18	15	102	ramesh	3	IEEE

subjectcode	subjectname
601	comp arch
604	financial
602	human manag

stidno	stidname	course	depno	depname	subjectcode
101	mary	Bsc	2	CA	601
102	reena	Bsc	2	CA	601
103	sudha	Bsc	2	CA	601
104	sasi	Bcom	3	HR	604
102	reena	Bsc	2	CA	604
101	mary	Bsc	2	CA	602
104	sasi	Bcom	3	HR	602

Figure 6. Result of the proposed system

**V.CONCLUSION AND FUTURE ENCHANGEMENTS**

This research work performed 1<sup>st</sup> normal form, 2<sup>nd</sup> normal form and 3<sup>rd</sup> normal form. These phases are used to normalize the database tables automatically. The status of the research work was completed with the 1<sup>st</sup> normal form. Going forward the normalization of 2<sup>nd</sup> normal form and 3<sup>rd</sup> normal form will be done. The future enhancements that can be done in system

are as follows normalization can be extended up to fifth normal form.

**REFERENCE**

- [1] Amir Hassan Bahmani, Mahmoud Naghibzadeh, Behnam Bahmani “An Automatic Database Normalization – Primary Key Generation”, IEEE(2008).
- [2] Sherry Verma , “Comparing manual and automatic normalization techniques for relational database”. IJREAS ,February 2012.
- [3] Abbasifard, Rahgozar, Pournemati “Using automated database reverse engineering for database integration”(2006).
- [4] Akehurst, Bordbar, Rodgers “Automatic normalization via meta modeling”.
- [5] Radha Krishna, Rajuvelpula, sravya “A web based relational database design tool to perform normalization”(2011).IEEE Journal.

G.Sunitha, Final Year MCA, Department of Computer Application, B. s. Abdur Rahman University.

Dr.A. Jaya , Professor , Department of Computer Application, B. S. Abdur Rahman University.